Pure and Applied Mathematics Journal
2013; 2(5): 162-168
Published online October 30, 2013 (http://www.sciencepublishinggroup.com/j/pamj)
doi: 10.11648/j.pamj.20130205.13

A decomposable computer oriented method for solving interval LP problems

Sharmin Afroz, M. Babul Hasan

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

Email address:
Sharmin_Afroz71@yahoo.com(S. Afroz)

To cite this article:
Sharmin Afroz, M. Babul Hasan. A Decomposable Computer Oriented Method for Solving Interval LP Problems. *Pure and Applied Mathematics Journal.* Vol. 2, No. 5, 2013, pp. 162-168. doi: 10.11648/j.pamj.20130205.13

Abstract: The purpose of this paper is to develop a computer oriented decomposition program for solving Interval Linear Programming (ILP) Problems. For this, we first analyze the existing methods for solving ILP problems. We also discuss the main stricter of Decomposable Interval programming (DIP) problems. Then a decomposable algorithm is analyzed for solving DIP problems. Using "Mathematica", we develop a computer oriented program for solving such problems. We present step by step illustration of a numerical example to demonstrate our technique.

Keywords: LP, ILP, DILP Computer Program

1. Introduction

Linear programming (LP) is a technique for determining on optimum schedule (such as maximizing profit or minimizing cost) of interdependent activities in view of the available resources. Programming is just another word for 'planning' and refers to the process of determining a particular plan of action from amongst 'linear' stands for indicating that all relationships involved in a particular problem are linear. Any LP consists of four parts: a set of decision variables, the parameters, the objectives function and a set of constraints. The constraints may be equalities or inequalities [1]. In particular, an LP can be written as

$$\begin{aligned} &\text{Maximize} \quad c^t x \\ &\text{Subject to (s.t.)} \quad Ax \le b \qquad (1) \\ &x \ge 0 \text{ [2].} \end{aligned}$$

General Form of ILP

An interval linear program or interval linear programming problem is any problem of the form [3]

(ILP): maximize c^tX

s.t. $b^- \le AX \le b^+$

where, $c = (c_i)$, $b^- = (b_j^-)$, $b^+ = (b_j^+)$,

and $A = (a_{ij})$(i=1,…,n;j=1,…,m) are given, with $b^- \le b^+$.

Let S be the set of points satisfying the constraints S={x∈ R^n: $b^- \le AX \le b^+$}.

A point x∈S is called a feasible solution of ILP problem if S≠ ϕ otherwise infeasible.

If (1) is bounded then it is equivalent to the ILP

Maximize c^tx

s.t. $-Me \le Ax \le b$

$0 \le x \le Me$

where, e is a vector of once and

M is a sufficiently large positive scalar [4].

There are a few methods for solving the ILP problems. Rober and Ben-Isreal [15] discussed a new iterative method for solving ILP in 1970. That method applies general LP and is shown to be a dual method with multiple substitution. Gunn and Anders [16] show that the simplex method for LP and ILP are identical which is shown in a comparison between simplex method for LP and ILP problems. Radimir Viher [18] develop an analogous theorem to solve ILP problems. Nakahara Sasaki and Gen [19] investigate a LP problem with interval coefficients and proposed a new concept of constraints based on probability. None of the above papers addressed the decomposition of ILP problems. In our paper, we will develop a decomposable technique for solving ILP problems.

Decomposable Interval Programming Problem (DILP)

An interval linear programming problem is a special type of interval problem [5]. A decomposable interval program (DILP) is any problem having the form [4]

(DILP): Maximize $c^t x$

s.t. $\bar{b}^- \le \bar{A}x \le \bar{b}^+$

$\hat{b}^- \le \hat{A}x \le \hat{b}^+$

Where $\bar{A}$ and $\hat{A}$ are nonsingular, $\bar{b}^- \le \bar{b}^+$, and $\hat{b}^- \le \hat{b}^+$.

In the next section, we will show how an ILP can be converted to DILP. Hence we will discuss an existing method to solve DILP. We will present our algorithm and corresponding coding using Mathematica in section IV. Using numerical example we will show that the results are same which are found using existing method and our algorithm.

The rest of the paper is organized as follows. In Section 2, conversion of ILP to DILP will be shown. For solving DILP a method related to the Dantzig-Wolfe decomposition principle will be discussed in section 3. In section 4, we will present our algorithm and computer technique to solve DILP. In section 5, we will conclude the paper.

2. Convertion of ILP to DILP

Conversion of ILP to DILP is shown in this section [6]. In general the general ILP form is given below [4]

(ILP): maximize c^tX

s.t. $b^- \le AX \le b^+$

Now let us consider ILP in the following form

(ILP): maximize c^tX

s.t.
$$b_1^- \le A_1X \le b_1^+$$
$$b_2^- \le A_2X \le b_2^+$$
$$b_3^- \le A_3X \le b_3^+$$

where, $A_1 \in R_r^{r\times r}$, $A_2 \in R_q^{q\times r}$ and $A_3 \in R^{(m-r-q)\times r}$. That is A_1 is a nonsingular submatrix of A, A_2 is any submatrix having full row rank whose rows are not in A_1 and A_3 is made up of the rows A not in A_1 or A_2. Note that q is not uniquely defined, but as we shall see later it is desirable to make q as large as possible (q=0 is always possible).

Clearly we can always choose

$b_4^- \le BX \le b_4^+$, a subset of the constraints (ILP) such that $\binom{A_2}{B} \in R_r^{r\times r}$.

Problem (ILP) is not changed by including some constraints more than once, so that ILP can be written as follows.

(ILP): maximize c^tX

s.t.
$$b_1^- \le A_1X \le b_1^+$$
$$b_3^- \le A_3X \le b_3^+$$
$$b_4^- \le BX \le b_4^+$$
$$b_2^- \le A_2X \le b_2^+$$

Finally, observe that x^* is an optimal solution to (ILP) iff (x^*, y^*) is an optimal solution to the following bounded problem which has the form

(DILP): maximize c^tX

$$\text{s.t. } \begin{pmatrix} b_1^- \\ 0 \end{pmatrix} \le \begin{pmatrix} A_1 & 0 \\ 0 & I_{m-r-q} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} b_1^+ \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} b_2^- \\ b_4^- \\ b_3^- \end{pmatrix} \le \begin{pmatrix} A_2 & 0 \\ B & 0 \\ A_3 & I_{m-r-q} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} b_2^+ \\ b_4^+ \\ b_3^+ \end{pmatrix}.$$

A_1, A_2 and A_3 can be identified by Gauss-Jordan elimination method. A procedure for identifying appropriate A_1, A_2, A_3 and B is demonstrated in the following example

Numerical Example 1

Transfer the following ILP problem to form DILP.

(DILP): maximize $x_1 + x_2$

$$\text{s.t. } \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \le \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 1 \\ -1 & -1 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}.$$

Solution

Applying Gauss-Jordan eliminations to A we obtain (pivots marked by asterisk):

$$\begin{pmatrix} 1^* & 1 \\ 2 & 2 \\ 0 & 1 \\ -1 & -1 \\ 2 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 1^* \\ 0 & 0 \\ 0 & -2 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Since rows 1 and 3 contained pivots we conclude that $A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ is a suitable nonsingular sub matrix of A. Rearranging A with matrix A_1 at the bottom we get $\begin{pmatrix} 2 & 2 \\ -1 & -1 \\ 2 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$.

To find next sub matrix we repeat the above procedure $\begin{pmatrix} 2* & 2 \\ -1 & -1 \\ 2 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & -2* \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$

We conclude that row 4 of A is linearly dependent on row 2 (since all its elements vanished after step 1) so that row 4 must fall in A_3. Furthermore since r=2 pivots were found before reaching the bottom r rows we conclude that B=0.Therefore, $A_2 = \begin{pmatrix} 2 & 2 \\ 2 & 0 \end{pmatrix}$ (rows 2 and 4 contained pivots),

$A_3 = (-1 \quad -1)$ and B=0.

The equivalent DILP is as follows

(DILP): maximize $x_1 + x_2$

$$\text{s.t. } \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix} \le \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y \end{pmatrix} \le \begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix} \le \begin{pmatrix} 2 & 2 & 0 \\ 2 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y \end{pmatrix} \le \begin{pmatrix} 3 \\ 6 \\ 5 \end{pmatrix}.$$

3. Existing Decomposition Method

In this section we discuss a method, for solving DILP which is related to the Dantzig-Wolfe decomposition principle [4].

The general form of DILP is

(DILP): Maximize $c^t x$

s.t. $\bar{b}^- \le \bar{A}x \le \bar{b}^+$

$\hat{b}^- \le \hat{A}x \le \hat{b}^+$

This may be written as

(DILP): Maximize $c^t x$

s.t. $\bar{x} = \hat{x}$

$\bar{b}^- \le \bar{A}\bar{x} \le \bar{b}^+$

$\hat{b}^- \le \hat{A}\hat{x} \le \hat{b}^+$.

Let $\bar{S}$={x$\in R^p$: $\bar{b}^- \le \bar{A}x \le \bar{b}^+$} and let $\bar{G}$ be the finite matrix whose columns are the extreme points of $\bar{S}$. Since $\bar{A}$ is nonsingular, $\bar{S}$ is a bounded polyhedron so that $\hat{x} \in \bar{S}$ iff $\bar{x} = \bar{G}\bar{v}, e^t\bar{v} = 1, \bar{v} \ge 0$(i.e. $\bar{x}$ is a convex combination of the extreme points of $\bar{S}$).An analogous result holds for $\hat{S}$ ={x $\in R^p$: $\hat{b}^- \le \hat{A}x \le \hat{b}^+$ }and the corresponding matrix $\hat{G}$ whose columns are the extreme points of $\hat{S}$.

Consequently, DILP may be written as

(DILP): maximize $c^t\bar{G}\bar{v}$

s.t. $\bar{G}\bar{v} - \hat{G}\hat{v} = 0$

$e^t\bar{v} = 1.$

$e^t\hat{v} = 1.$

$\bar{v}$,$\hat{v} \ge 0.$

This problem has standard linear programming form except that the columns of $\bar{G}$ and $\hat{G}$ is not immediately known. Indeed, we shall solve the constraints using the simplex algorithm with a special technique for generating the columns of $\bar{G}$ and $\hat{G}$ one at a time as needed.

Suppose that a basic feasible solution to the constraints is in hand with simplex multipliliers

$(\pi_1, \dots, \pi_p, \sigma_1, \sigma_2) = (\pi, \sigma_1, \sigma_2)$.Let the columns of $\bar{G}$ and $\hat{G}$ be denoted by $\bar{g}_i$ (i=1,…. , $\bar{N}$) and $\hat{g}_i$(i=1, … ,$\hat{N}$) respectively. A vector$\begin{pmatrix}\bar{g}_i\\1\\0\end{pmatrix}$ or $\begin{pmatrix}-\hat{g}_i\\0\\1\end{pmatrix}$ can enter the basis if

$(\pi, \sigma_1, \sigma_2)\begin{pmatrix}\bar{g}_i\\1\\0\end{pmatrix}$ - $c^t\bar{g}_i = (\pi - c^t)\bar{g}_i + \sigma_1 < 0$

or $(\pi, \sigma_1, \sigma_2)\begin{pmatrix}-\hat{g}_i\\0\\1\end{pmatrix}$ = - $\pi\hat{g}_i + \sigma_2$<0 respectively.

The smallest relative cost λ is defined as

λ= min $\{min_{\bar{g}_i}((\pi - c^t)\bar{g}_i + \sigma_1), min_{\hat{g}_i}(-\pi\hat{g}_i + \sigma_2)\}$ (2)

Following the standard simplex procedure we bring into the vector with the smallest relative cost. If $\lambda \ge 0$,the solution is optimal.

Thus we extreme points $\bar{g}^*$ and $\hat{g}^*$ such that $(\pi - c^t)\bar{g}^* = min_{\bar{g}_i}((\pi - c^t)\bar{g}_i)$ and

$-\pi\hat{g}^* = min_{\hat{g}_i}(-\pi\hat{g}_i)$.But

$min_{\bar{g}_i}((\pi - c^t)\bar{g}_i) = min_{x\in\bar{S}}((\pi - c^t)x)$ so that

$\bar{g}^* = \sum_{i\in\bar{I}_+} \bar{b}_i^+\bar{t}_i + \sum_{i\in\bar{I}_-} \bar{b}_i^-\,\bar{t}_i$.

Where $\bar{A}^{-1}$= $(\bar{t}_1, \dots, \bar{t}_p)$ and

$\bar{I}_{-,+} = \{i: (\pi - c^t)\bar{t}_i \ge, < 0\}$.

(i.e. $\bar{g}^*$ is an extreme point solution of the sub problem $min_{x\in\bar{S}}((\pi - c^t)x)$).

Likewise $\hat{g}^*$= $\sum_{i\in\hat{I}_+} \hat{b}_i^+\hat{t}_i + \sum_{i\in\hat{I}_-} \hat{b}_i^-\,\hat{t}_i$

Where, $\hat{A}^{-1}$ = $(\hat{t}_1, \dots, \hat{t}_p)$ and

$\hat{I}_{-,+}$ ={i: -$\pi\hat{t}_i \ge, < 0$}.

If λ<0 either $\begin{pmatrix}\bar{g}^*\\1\\0\end{pmatrix}$ or $\begin{pmatrix}\hat{g}^*\\0\\1\end{pmatrix}$ enter the basis depending on which has the lowest relative cost. The simplex iteration is then completed to obtain a better basic feasible solution and the entire procedure is repeated.

If λ≥ 0 the present solution, say $(\bar{v}^*, \hat{v}^*)$ to $\bar{G}\bar{v} - \hat{G}\hat{v} = 0$ is optimal. An optimal solution to (DILP) is then

$$x^* = \sum_{i\in\Omega_1} \bar{g}_i\bar{v}_i^* = \sum_{i\in\Omega_2} \hat{g}_i\hat{v}_i^* \quad (3)$$

where, $\Omega_1 = \{i: \bar{v}_i > 0\}$ and Ω_2= {i:$\hat{v}_i > 0$}.

An initial basic feasible solution can be found using artificial variables. One approach is to being with the enlarged problem

Maximize $c^t\bar{G}\bar{v} - Me^tz$

$$\text{s.t.} \begin{pmatrix}\bar{G} & -\hat{G} & \\ e^t & 0 & I_{p+2}\\ 0 & e^t & \end{pmatrix}\begin{pmatrix}\bar{v}\\\hat{v}\\z\end{pmatrix} = \begin{pmatrix}0_{p\times1}\\1\\1\end{pmatrix} \quad (4)$$

$\bar{v}$, $\hat{v}$,z ≥0

Where M is a sufficiently large scalar, using $\{e_1, \dots, e_{p+2}\}$ as the starting basis. If λ≥ 0 while some of the artificial variable are still in the basis at a non-zero level, problem (DILP) has no feasible solution.

A slightly different approach for obtaining an initial basic feasible solution to the constraint is to find any extreme points of $\bar{S}$ and $\hat{S}$. If we call these points $\bar{g}$ and $\hat{g}$ respectively, then

$$\begin{pmatrix}\bar{g}_1\\1\\0\end{pmatrix}, \begin{pmatrix}-\hat{g}_1\\0\\1\end{pmatrix}, \pm\begin{pmatrix}e_1\\0\\0\end{pmatrix}, \dots, \pm\begin{pmatrix}e_p\\0\\0\end{pmatrix} \quad (5)$$

is a suitable starting basis to the equivalent problem

Maximize $c^t\bar{G}\bar{v} - Me^tz$

$$\text{s.t}\begin{pmatrix}\bar{G} & -\hat{G} & \\ e^t & 0 & I_{p+2}\\ 0 & e^t & \end{pmatrix}\begin{pmatrix}\bar{v}\\\hat{v}\\z\end{pmatrix} = \begin{pmatrix}0_{p\times1}\\1\\1\end{pmatrix} \quad (6)$$

$\bar{v}$, $\hat{v}$,z ≥0

The sign in front of $\begin{pmatrix}e_i\\0\\0\end{pmatrix}$ in (5) will depend on the sign of $(\bar{g}_{1i} - \hat{g}_{1i})$ and has purposely been left ambiguous to simplify notation.

Numerical Example 2

Find the optimal solution of the ILP problem
(ILP): maximize $x_1 + 2x_2$
s.t. $0 \le x_1 \le 6$
$0 \le x_2 \le 8$
$2 \le x_1 + x_2 \le 6$
$-9 \le -3x_1 + x_2 \le 9.$

Solution

Given that
(ILP): maximize $x_1 + 2x_2$
s.t. $0 \le x_1 \le 6$
$0 \le x_2 \le 8$
$2 \le x_1 + x_2 \le 6$
$-9 \le -3x_1 + x_2 \le 9.$

A decomposable interval problem has the form,
(DILP): maximize $c^t x$
s.t. $\bar{b}^- \le \bar{A}x \le \bar{b}^+$
$\hat{b}^- \le \hat{A}x \le \hat{b}^+.$

The given problem is a decomposable interval problem having

$\bar{b}^- = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\bar{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,
$\bar{A}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\bar{b}^+ = \begin{pmatrix} 6 \\ 8 \end{pmatrix}$.
$\hat{b}^- = \begin{pmatrix} 2 \\ -9 \end{pmatrix}$, $\hat{A} = \begin{pmatrix} 1 & 1 \\ -3 & 1 \end{pmatrix}$,
$\hat{A}^{-1} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix}$, $\hat{b}^+ = \begin{pmatrix} 6 \\ 9 \end{pmatrix}$.

Now $\bar{S} = \{x \in R^2 : 0 \le x_1 \le 6, 0 \le x_2 \le 8\}$
$\hat{S} = \{x \in R^2 : 2 \le x_1 + x_2 \le 6, -9 \le -3x_1 + x_2 \le 9\}$.

The equivalent problem of the given problem is
Maximize $c^t \bar{G}\bar{v} - Me^t z$

s.t. $\begin{pmatrix} \bar{G} & -\hat{G} & I_{p+2} \\ e^t & 0 & \\ 0 & e^t & \end{pmatrix} \begin{pmatrix} \bar{v} \\ \hat{v} \\ z \end{pmatrix} = \begin{pmatrix} 0_{p\times 1} \\ 1 \\ 1 \end{pmatrix}$

$\bar{v}, \hat{v}, z \ge 0$

Where $c^t = \begin{pmatrix} 1 \\ 2 \end{pmatrix}^t$

$$\bar{G}\bar{v} = (\sum_{i=1}^{\bar{N}} \bar{g}_i \, \bar{v}_i)$$

$Z = (z_1, z_2)$

$\bar{G} = \begin{pmatrix} \bar{g}_1 & \dots & \bar{g}_{\bar{N}} \\ 1 & \dots & 1 \\ 0 & \dots & 0 \end{pmatrix}$; $\hat{G} = \begin{pmatrix} -\hat{g}_i & \dots & -\hat{g}_{\bar{N}} \\ 0 & \dots & 0 \\ 1 & \dots & 1 \end{pmatrix}$; $I_{p+2} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$

Iteration-0:

According to the problem

$$\bar{g}_1 = \bar{A}^{-1}\bar{b} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 8 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \end{pmatrix}$$

$$\hat{g}_1 = \hat{A}^{-1}\hat{b} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 6 \\ 9 \end{pmatrix} = \begin{pmatrix} -\frac{3}{4} \\ \frac{27}{4} \end{pmatrix}.$$

The optimal function is

Maximize $\begin{pmatrix} 1 \\ 2 \end{pmatrix}^t (\sum_{i=1}^{\bar{N}} \bar{g}_i \, \bar{v}_i) - M(z_1 + z_2)$

$= \begin{pmatrix} 1 \\ 2 \end{pmatrix}^t \bar{g}_1 \bar{v}_1 + 0.\hat{v} - M(z_1 + z_2)$

$= (1 \quad 2) \begin{pmatrix} 6 \\ 8 \end{pmatrix} \bar{v}_1 + 0.\hat{v} - M(z_1 + z_2)$

$= 22\bar{v}_1 + 0.\hat{v} - M(z_1 + z_2)$

Subject to

$$\begin{pmatrix} 6 & \frac{3}{4} & -1 & 0 \\ 8 & -\frac{27}{4} & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{v} \\ \hat{v} \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$\bar{v}, \hat{v}, z_1, z_2 \ge 0.$

Now the columns of

$B = \begin{pmatrix} 6 & \frac{3}{4} & -1 & 0 \\ 8 & -\frac{27}{4} & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$ form a basis.

Now inverse of B is

$$B^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 6 & \frac{3}{4} \\ 0 & -1 & 8 & -\frac{27}{4} \end{pmatrix}$$

Therefore the solution is

$$\begin{pmatrix} \bar{v}_1 \\ \hat{v}_1 \\ z_1 \\ z_2 \end{pmatrix} = B^{-1} \times \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$\bar{v}_1 = 1, \quad \hat{v}_1 = 1, \quad z_1 = \frac{27}{4}, \quad z_2 = \frac{5}{4}$

Here simplex multipliers are
$(\pi, \sigma_1, \sigma_2)$ = (22, 0,-M,-M)B^{-1}
= (M, M, 22-14M, 6M)
∴ π= (M, M), $\sigma_1 = 22 - 14M$, $\sigma_2 = 6M$

Iteration-1:

π-c^t= (M, M)- $\begin{pmatrix} 1 \\ 2 \end{pmatrix}^t$
= (M, M)- (1, 2)
= (M-1, M-2)

Since

(π- c^t) $\bar{t}_1 = (M-1 \quad M-2) \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ = M-1>0 and (π-c^t)$\bar{t}_2 = (M-1 \quad M-2) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ =M-2>0

So, $\bar{g}^* = 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Again, $-\pi\ \hat{t}_i = (-M \quad -M)\begin{pmatrix}\frac{1}{4}\\ \frac{3}{4}\end{pmatrix} = -M < 0$ and $-\pi\hat{t}_2 = (-M \quad -M)\begin{pmatrix}\frac{-1}{4}\\ \frac{1}{4}\end{pmatrix} = 0$

So, $\hat{g}^* = 6\begin{pmatrix}\frac{1}{4}\\ \frac{3}{4}\end{pmatrix} - 9\begin{pmatrix}\frac{-1}{4}\\ \frac{1}{4}\end{pmatrix} = \begin{pmatrix}\frac{15}{4}\\ \frac{9}{4}\end{pmatrix}$

The corresponding relative costs are

$(\pi\text{-}c^t)\bar{g}^* + \sigma_1 = -14M + 22 < 0$

$-\pi\ \hat{g}^* + \sigma_2 = (-M \quad -M)\begin{pmatrix}\frac{15}{4}\\ \frac{9}{4}\end{pmatrix} + 6M = -\frac{15M}{4} - \frac{9M}{4} + 6M = \frac{-24M+24M}{4} = 0$

The smallest relative cost

$\lambda = \min\{min_{\bar{g}_i}((\pi - c^t)\bar{g}^* + \sigma_1), min_{\hat{g}_i}(-\pi\hat{g}^* + \sigma_2)\}$=min {22-14M, 0 }

$=22 - 14M < 0$

Thus $\begin{pmatrix}\bar{g}_2\\ 1\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 1\\ 0\end{pmatrix}$ enters the basis.

$$\therefore B^{-1}\begin{pmatrix}\bar{g}_2\\ 1\\ 0\end{pmatrix} = \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -1 & 0 & 6 & \frac{3}{4}\\ 0 & -1 & 8 & -\frac{27}{4}\end{pmatrix}\begin{pmatrix}0\\ 0\\ 1\\ 0\end{pmatrix} = \begin{pmatrix}1\\ 0\\ 6\\ 8\end{pmatrix}.$$

Applying the minimum ratio rule the ratios are $(\infty,\infty,1/6,1/8)$. Hence column 4, $\begin{pmatrix}0\\ -1\\ 0\\ 0\end{pmatrix}$ replaced by $\begin{pmatrix}0\\ 0\\ 1\\ 0\end{pmatrix}$.

The new basis vectors are the columns of

$$B = \begin{pmatrix}6 & \frac{3}{4} & -1 & 0\\ 8 & -\frac{27}{4} & 0 & 0\\ 1 & 0 & 0 & 1\\ 0 & 1 & 0 & 0\end{pmatrix}$$

Inverse of the matrix is

$$B^{-1} = \begin{pmatrix}0 & \frac{1}{8} & 0 & \frac{27}{32}\\ 0 & 0 & 0 & 1\\ -1 & \frac{3}{4} & 0 & \frac{93}{16}\\ 0 & \frac{-1}{8} & 1 & \frac{-27}{32}\end{pmatrix}.$$

Similarly after the iterations we get the solution is

$$\begin{pmatrix}\bar{v}_1\\ \hat{v}_1\\ z_1\\ \bar{v}_2\end{pmatrix} = B^{-1}\begin{pmatrix}0\\ 0\\ 1\\ 1\end{pmatrix} = \begin{pmatrix}\frac{27}{32}\\ 1\\ \frac{93}{16}\\ \frac{5}{32}\end{pmatrix}$$

$\therefore \bar{v}_1 = \frac{27}{32}, \hat{v}_1 = 1, z_1 = \frac{93}{16}, \bar{v}_2 = \frac{5}{32}$

Simplex multipliers are

$(\pi, \sigma_1, \sigma_2)$ = (22, 0, -M, 0)

$$B^{-1} = (22, 0, -M, 0)\begin{pmatrix}0 & \frac{1}{8} & 0 & \frac{27}{32}\\ 0 & 0 & 0 & 1\\ -1 & \frac{3}{4} & 0 & \frac{93}{16}\\ 0 & \frac{-1}{8} & 1 & \frac{-27}{32}\end{pmatrix}$$

$= (M, \frac{22}{8} - \frac{3}{4}M, 0, \frac{297}{16} - \frac{93}{16}M)$.

The smallest relative cost $\lambda = \min\{min_{\bar{g}_i}((\pi - c^t)\bar{g}^* + \sigma_1), min_{\hat{g}_i}(-\pi\hat{g}^* + \sigma_2)\}$ =min {0, 0}= 0

So the optimal basic feasible solution is

$x^* = \sum \bar{g}_i\bar{v}_i^* = \frac{3}{4}\binom{0}{8} + \frac{1}{4}\binom{0}{0} = \binom{0}{6}$ Or $x^* = \sum \hat{g}_i\hat{v}_i^* = \frac{1}{6}\begin{pmatrix}\frac{15}{4}\\ \frac{9}{4}\end{pmatrix} + \frac{5}{6}\begin{pmatrix}\frac{-3}{4}\\ \frac{27}{4}\end{pmatrix} = \binom{0}{6}$

The optimum value is $x_1 + 2x_2$= 0 + 2.6 = 12.

4. Algorithm for Solving ILP

In this section, we will present our algorithm for solving ILP.

Algorithm

An algorithm for the technique is given below

Step-1: First we will enter the basis matrix B and constant matrix, a row matrix for simplex multipliers, two matrixes $\bar{A}$ *and* $\hat{A}$ for constraints. We will find a solution.

Step-2: Then compute $\bar{g}^*$ and $\hat{g}^*$.

Step-3: Find relative cost for each. After that find minimum relative cost λ. If the relative cost is greater or equal to zero the solution is optimal. Otherwise, any column of the basis matrix will change by the column $\begin{pmatrix}\bar{g}_i\\ 1\\ 0\end{pmatrix}$ or $\begin{pmatrix}-\hat{g}_i\\ 0\\ 1\end{pmatrix}$ depending on the minimum relative cost.

Step-4: Using the minimum ratio rule the corresponding column of the variable is changed by the column found before.

Computer Code for Solving ILP

In this section, we will develop a computer code using computer algebra Mathematica [12]. This is given as follows.

Itration-1

```
<< LinearAlgebra`MatrixManipulation`
```

$$B = \begin{pmatrix} 6 & \frac{3}{4} & -1 & 0 \\ 8 & \frac{-27}{4} & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix};$$

$$co = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix};$$

$$c = \begin{pmatrix} 1 \\ 2 \end{pmatrix};$$

$$\bar{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};$$

$$\hat{A} = \begin{pmatrix} 1 & 1 \\ -3 & 1 \end{pmatrix};$$

```
A1 = Inverse[Â];
A2 = Inverse[Ā];
t̄1 = TakeColumns[A2, 1];
t̄2 = TakeColumns[A2, -1];
t̂1 = TakeColumns[A1, 1];
t̂2 = TakeColumns[A1, -1];
iB = Inverse[B];
Print["the simplex multipliers (π σ1 σ2)=", sm = (22 0 -M -M).iB]

Print["π =", TakeColumns[sm, 2] // MatrixForm]
Print["The solution is (v̄1 ,v̂1,z1,z2)=", ls = LinearSolve[B, co]]

p = (TakeColumns[sm, 2] - Transpose[c]);
q = p.t̄1;
m = q /. M → 1000;
a = 0 TakeColumns[A2, 1];
b = 6 TakeColumns[A2, 1];
g1 = If[m[[1, 1]] ≥ 0, a, b];
q1 = p.t̄2;
m1 = q1 /. M → 1000;
a1 = 0 TakeColumns[A2, -1];
b1 = 8 TakeColumns[A2, -1];
g2 = If[m1[[1, 1]] ≥ 0, a1, b1];
ḡ* = g1 + g2

s = -TakeColumns[sm, 2].t̂1;
n = s /. M → 1000;
aa = 2 TakeColumns[A1, 1];
bb = 6 TakeColumns[A1, 1];
gg1 = If[n[[1, 1]] ≥ 0, aa, bb];
t = -TakeColumns[sm, 2].t̂2;
n1 = t /. M → 1000;
aa1 = -9 TakeColumns[A1, -1];
bb1 = 9 TakeColumns[A1, -1];
gg2 = If[n1[[1, 1]] ≥ 0, aa1, bb1];
ĝ* = gg1 + gg2

rc1 = p.(g1 + g2) + sm[[1, 3]];
rc2 = -TakeColumns[sm, 2].(gg1 + gg2) + sm[[1, 4]];
r1 = rc1 /. M → 1000;
r2 = rc2 /. M → 1000;
λ = Min[r1, r2];
lst1 = (ḡ*, 1, 0);
lst11 = Flatten[lst1];
lst2 = (-ĝ*, 0, 1);
lst22 = Flatten[lst2];
If[λ ≥ 0, Print["the solution is optimal"],
 If[λ == r1[[1, 1]], Print["(ḡ*, 1, 0)=", lst11], Print["(-ĝ*, 0, 1)=", lst22]]]

b1 = iB.lst11;
ra1 = ls[[1, 1]] / b1[[1]];
ra2 = ls[[2, 1]] / b1[[2]];
ra3 = ls[[3, 1]] / b1[[3]];
ra4 = ls[[4, 1]] / b1[[4]];
Lst[ra1, ra2, ra3, ra4]
```

Now we get the solution as follows

```
the simplex multipliers (π σ1 σ2)={{M, M, 22 - 14 M, 6 M}}

π = ( M   M )

The solution is (v̄1 ,v̂1,z1,z2)={{1}, {1}, {27/4}, {5/4}}
{{0}, {0}}

{{15/4}, {9/4}}          (ḡ*, 1, 0)={0, 0, 1, 0}

Out[55]= Lst[1, ComplexInfinity, 9/8, 5/32]
```

Now the minimum ratio is 5/32. So column 4 is replaced by $\begin{pmatrix} \bar{g}_2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$.

Similarly, after some iterations we get the solution given below

the simplex multipliers $(\pi\ \sigma_1\ \sigma_2)$={{2, 2, 0, 12}}

π =(2 2)

The solution is $(\overline{v_1}, \hat{v}_1, z_1, z_2)=\{\{\frac{3}{4}\}, \{\frac{5}{6}\}, \{\frac{1}{6}\}, \{\frac{1}{4}\}\}$

{{0}, {0}}

$\{\{\frac{15}{4}\}, \{\frac{9}{4}\}\}$

the solution is optimal

So that the optimal basic feasible solution is

$x^* = \sum \bar{g}_i \bar{v}_i^* = \frac{3}{4}\binom{0}{8} + \frac{1}{4}\binom{0}{0} = \binom{0}{6}$ Or

$$x^* = \sum \hat{g}_i \hat{v}_i^* = \frac{1}{6}\begin{pmatrix} \frac{15}{4} \\ \frac{9}{4} \end{pmatrix} + \frac{5}{6}\begin{pmatrix} \frac{-3}{4} \\ \frac{27}{4} \end{pmatrix} = \binom{0}{6}$$

The optimum value is $x_1 + 2x_2$
=0 + 2.6
= 12

5. Conclusion

In this paper, we have analyzed a decomposable method for solving interval linear programming problems. We then developed a computer technique using Mathematica which reduces our time and effort for solving such problems. In the decomposable process, ILP has to be converted into a decomposable ILP (DILP) which is used to solve ILP problems. But in this method a lot of calculations have to be done which is time consuming and mistakes can be occurred. Our computer technique was developed according to the DILP method and it minimizes these difficulties. So our computer technique is an effective process for solving ILP after converting it into a DILP. Numerical examples are illustrated to demonstrate our technique.

References

[1] Ravindran, Philips & Solberg (2000), "Operations Research", John Wiley and Sons, Second Edition, New York, U.S.A.

[2] Ganesh Chandra Ray, D., Md. Elias Hossain (2008), "Linear Programming," 3rd edition, Titas Publications, Dhaka-1100.

[3] Charners, A. Frieda Granot and F. Philips (1977), " *An Algorithm for Solving Interval Linear Programming Problems* ", Operation Research, Vol. 25, No.4, pp. 688-695.

[4] Ben-Isreal, A. and P.D. Robers (1970), " *A Decomposable Method for Interval Linear Programming* ", Management Science, Vol. 16, No.5, pp. 374-387.

[5] Dantzig, G.B. and P. Wolfe (1961), "*The Decomposition Algorithm for Linear Programming*", Econometrica, Vol. 29, No.4.

[6] Sweeny, D.J. and R.A. Murphy (1979), "*A Method of Decomposition for Integer Programs*", Operations Research, Vol. 27, No.6, pp. 1128-1141.

[7] Hasan, M.B. and J.F. Raffensperger (2007), "*A Decomposition Based Pricing Model for Solving a Large-Scale MILP Model for an Integrated Fishery*", Hindawi Publishing Corporation, Journal of Applied Mathematics and Decision Sciences, Vol. 2007, Article ID 56404, 10 pages.

[8] Winston, W.L. (1994), "*Linear Programming:Applications and Algorithm*", Duxbury press, Belmont, California, U.S.A.

[9] Dantzig, G.B. (1963), "*Linear Programming and Extensions*", Princeton University Press, Princeton, U.S.A..

[10] A B M Rezaul Karim, B M Ikramul Haque, Anisur Rahman & Muhammad Mofisur Rahman (2006), "*Linear Programming"*, First Edition

[11] Robert Fourer, David M.Gay & Brian W.Kernighan, "*A Modeling Language for Mathematical Programming"*, Secondt Edition.

[12] Don, E. (2000), "*Theory and Problems of Mathematica*", Schaum's Outline Series, Mc. GRAW-HILL

[13] Zangwill, W.i. (1967), " *A Decomposable Non- Linear Programming Approach*", Operations Research, Vol. 15, No.6, pp. 1068-1087.

[14] Sanders, J.L. (1965), " *A Nonlinear Decomposable Principle* ", Operation Research, Vol. 13, No.2, pp. 266-271.

[15] Rober P. D. and A. Ben-Isreal (1970) , " *A Suboptimization Method or Interval Linear Programming: A New Method for Linear Programming*", Linear Algebra and Its Applications 3, pp. 383-405.

[16] Gunn E. A. and G. J. Anders (1981), "*A Comparison of Interval Linear Programming with Simplex Method*", Linear Algebra And Its Applications, 38, pp. 149-159.

[17] Oliver Aberth (1997), " *The Solution of Linear Interval Equations by a Linear Programming Method*", Linear Algebra and Its Applications 259,pp. 271-279.

[18] Radimir Viher (2003), " *Interval Method for Interval Linear Program*", Mathematical Communications 8, pp. 23-33.

[19] Nakahara Y., M. Sasaki and M. Gen (1992), " *On the Linear Programming Problems withInterval Coefficients*", Computer and Industrial Engineering, Vol. 23, pp. 301-304.